R4. Information Systems

UNDERSTANDING ITS/CVO TECHNOLOGY APPLICATIONS
Reference Manual

MODULE 4 - INFORMATION SYSTEMS

References

1) *Introduction to Information Systems,* James A. Obrien, Irwin/McGraw-Hill, 1997, ISBN 0-256-20937-5.
2) *System Analysis and Design Methods,* Jeffrey L. Whitten & Lonnie Bentley, Irwin/McGraw-Hill, 1998, ISBN 0-256-19906-X.
3) *Principles and Guidelines in Software User Interface Designs*, Deborah J. Mayhew, Prentice Hall,1992, ISBN 0-13-721929-6.
4) *IEEE Std 1074-1995,* IEEE Standard for Developing Software Life Cycle Processes.
5) *IEEE Std 1074.1-1995,* IEEE Guide for Developing Software Life Cycle Processes.
6) *Software Engineering*, IEEE Standards Collection, 1997 Edition.
7) *The Software Development Project,* Phillip Bruce and Sam M. Pederson, John Wiley & Sons, 1982, ISBN 0-471-06269-3.
8) *Configuration Management for Software,* Stephen B. Compton and Guy R. Conner, Van Nostrand Reinhold, 1994, ISBN 0-442-01746-4.
9) *Software Engineering Standards*, A User's Roadmap, James W. Moore, IEEE Computer Society, 1997, ISBN 0-8186-8008-3.

Software Life Cycle Models

A few definitions for software life cycle models from IEEE Std 610.12-1990, the IEEE Standard Glossary of Software Engineering Terminology.

- **Waterfall Model**. A model of the software development process in which the constituent activities, typically in a concept phase, requirements phase, design phase, implementation phase, test phase, and installation and checkout phase, are performed in that order, possibly with overlap but with little or no iteration.
- **Incremental Development**. A software development technique in which requirements definition, design, implementation, and testing occur in an overlapping, iterative (rather than sequential) manner, resulting in incremental completion of the overall software product.
- **Rapid Prototyping**. A type of prototyping in which emphasis is placed on developing prototypes early in the development process to permit early feedback and analysis in support of the development process.
- **Prototyping**. A hardware and software development technique in which a preliminary version of part or all of the hardware or software is developed to permit user feedback, determine feasibility, or investigate timing or other issues in support of the development process.
- **Spiral Model**. A model of the software development process in which the constituent activities, typically requirements analysis, preliminary and detailed design, coding, integration, and testing, are performed iteratively until the software is complete.

AN EFFECTIVE APPROACH TO SYSTEM INTEGRATION: A COMPREHENSIVE CHECKLIST

7th Annual Symposium of the
International Council On Systems Engineering (INCOSE)
Los Angeles, California

6 August 1997

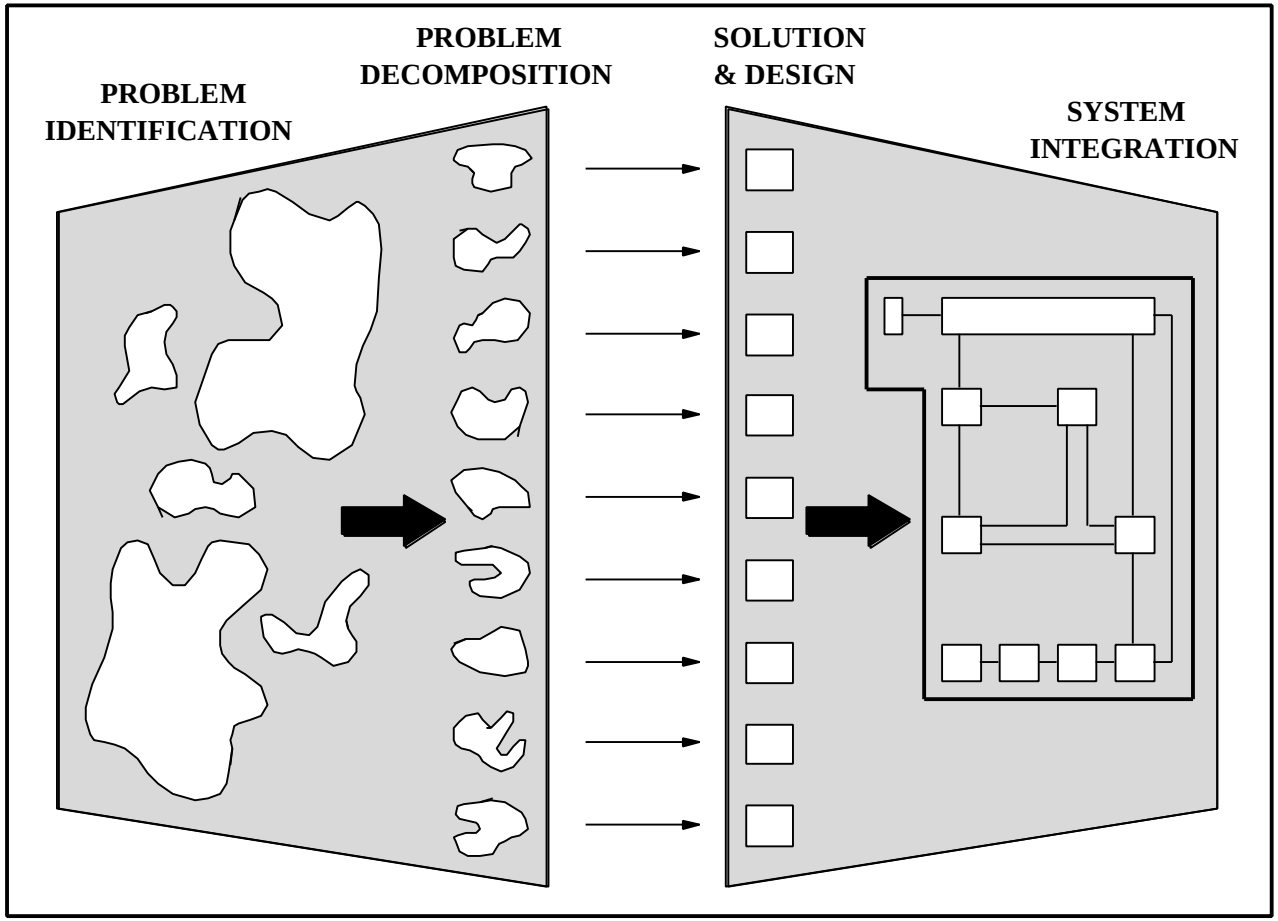
Scott A. Hyer
The Johns Hopkins University
Applied Physics Laboratory
Laurel, Maryland

Presentation Approach

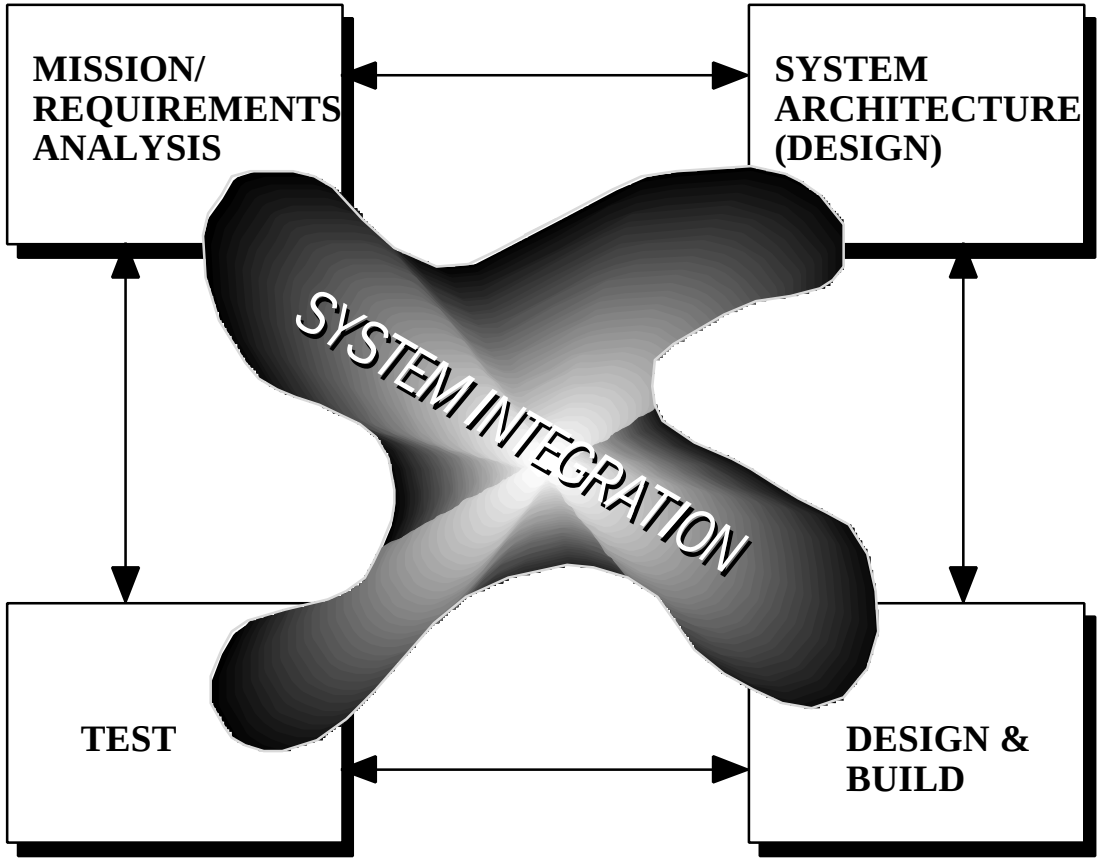
- System Integration Origins and Basics
- Different Perspectives- Good Practices
  - The Experts: Rechtin, Grady, Forsberg/Mooz, Martin
  - INCOSE SE Process Activities "How To" Guide
  - SECAM
- The Checklist

The Need for System Integration

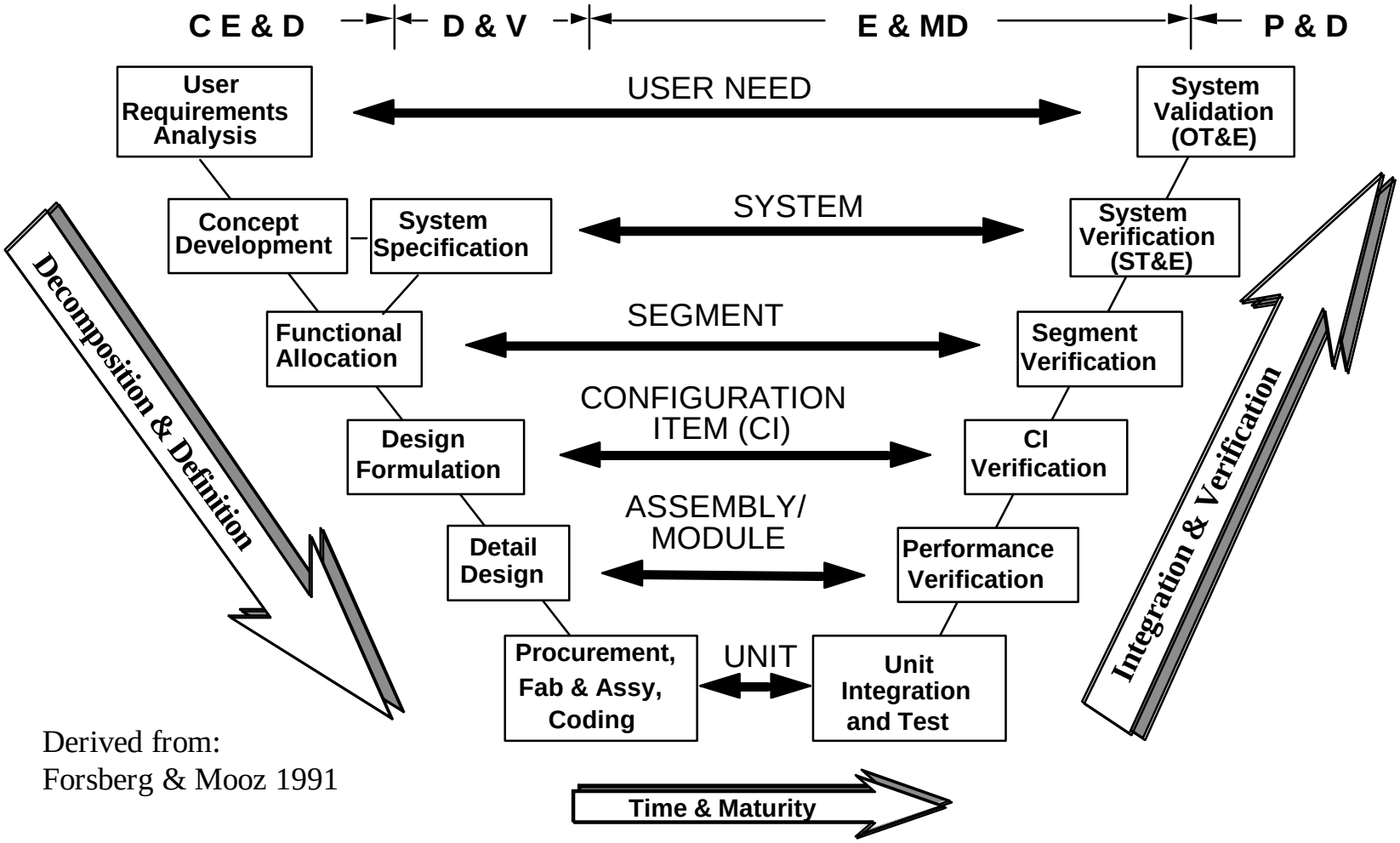
"Relationships among the elements are what give systems their added value" - Rechtin 1991



Where is System Integration in the SE Process Flow ?

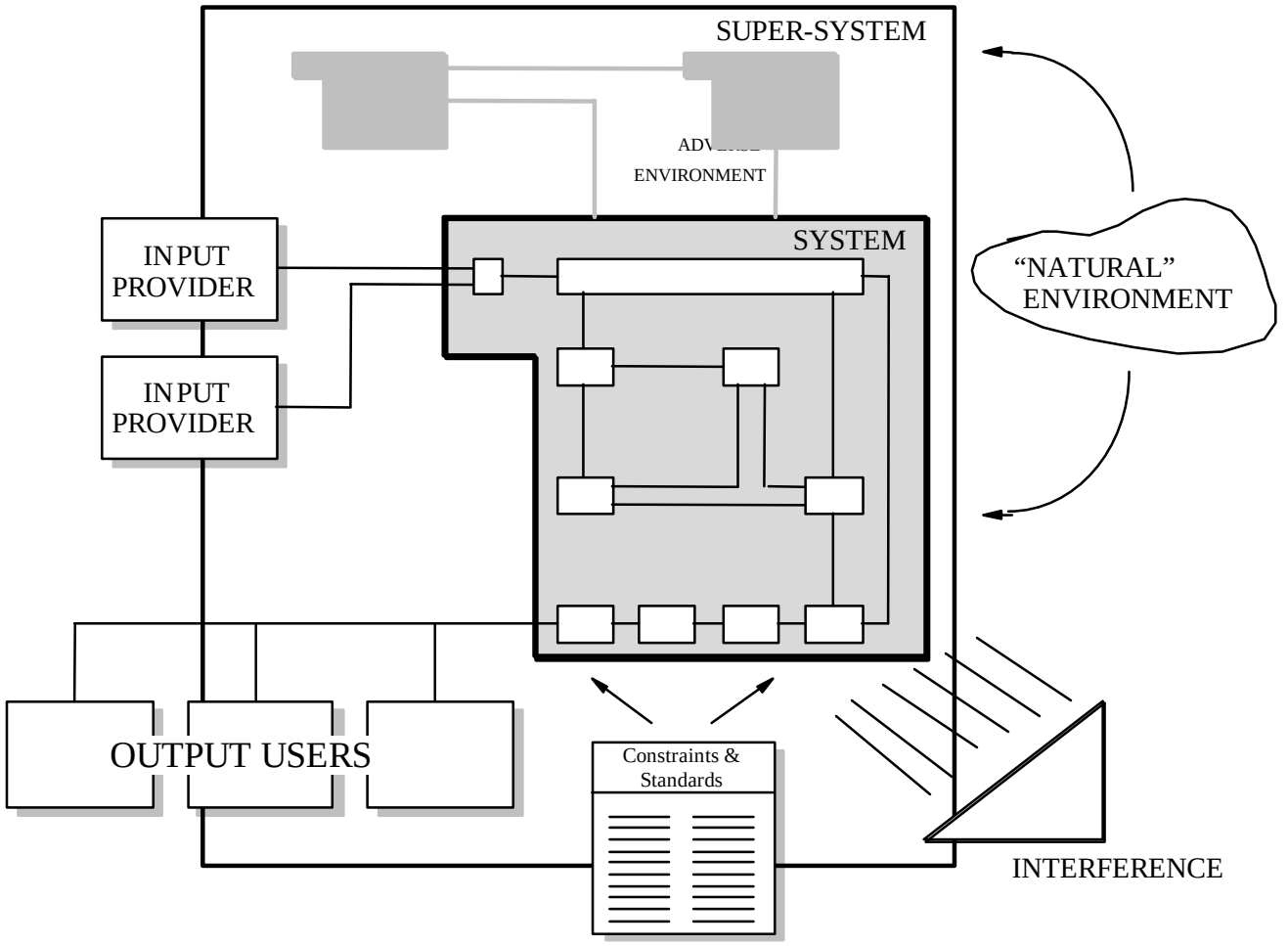


Project Cycle "V" Chart



Derived from:
Forsberg & Mooz 1991

Remember the “Big Picture”



SECAM* Perspective

Level 1	Level 2	Level 3	Level 4	Level 5
Performed	**Managed**	**Defined**	**Measured**	**Optimizing**
IS INTEGRATION DONE AT LEAST INFORMALLY ?	**IS A FOCUSED RESPONSIBILITY TAKEN FOR INTEGRATION ?**	**IS THERE A DEFINED INTEGRATION PROCESS ?**	**CAN PROBLEMS WITH THE INTEGRATION PROCESS BE DETECTED ?**	**IS THE INTEGRATION PROCESS BEING IMPROVED BY EXPERIENCE ?**

* Systems Engineering Capability Assessment Model, Version 1.50, June 1996, INCOSE CAWG

System Integration Checklist

Major Activities To Be Tailored

1. Define Context of Effort
2. Define Mission and Users
3. Define Operating Environment
4. Analyze Interactions
5. Prepare ICDs
6. Evaluate User Interfaces
7. Coordinate with Test Plans
8. Evaluate System Performance
9. Conflict Resolution

SYSTEM INTEGRATION CHECKLIST (1 of 3)

TASK/ACTIVITY	*TOOLS & PRODUCTS*
1. Define Context of Effort Write SOW to include all tasks Define "Internal" and "External" Bound effort to avoid over-extension	Context Diagram Statement of Work
2. Define Mission and Users Study existing information, know key players, reference existing systems and practices Understand system's purpose, user's perspective, and existing infrastructure	Concept of Ops/ORD Distributions List of Constraints
3. Define Operating Environment Enhance documentation, understand driving conditions Understand everything that can adversely impact the system	Environmental Descriptions External Drivers

SYSTEM INTEGRATION CHECKLIST (2 of 3)

TASK/ACTIVITY	*TOOLS & PRODUCTS*
4. Analyze Interactions Tabulate relationships which may be sources of risk (include requirements, components, ...) Identify interfaces, eliminate non-problems	N^2 Chart Risk Mitigation Plan
5. Prepare ICDs Define interface details and get concurrence Record agreements on interface designs	Interface Control Documents (ICDs)
6. Evaluate User Interfaces Involve User agents to analyze designs "on the fly" Involve users, get reactions, iterate	Mockups User Feedback/Interviews

SYSTEM INTEGRATION CHECKLIST (3 of 3)

TASK/ACTIVITY	***TOOLS & PRODUCTS***
7. <u>Coordinate with Test Plans</u> Ensure efforts mesh with test plan & schedule What needs to be connected ? When do things need to be ready ?	Test Plan and Descriptions
8. <u>Evaluate System Performance</u> Identify MOEs that are at risk Predict performance (re: interactions) Trade off candidate solutions	MOEs System Simulation
9. <u>Conflict Resolution</u> Communicate with the project team Keep up with the latest issues (keep AI list) Always have a contingency plan	Action Item (AI) List System Simulation

The Final Word ...

- System Integration starts at the beginning:
  - Know the system's boundaries, mission, users, and operating environment
  - Establish a mindset through requirements, analysis, design, and test activities
- System Integration acts as the "glue":
  - Work the interactions (interfaces, interference, etc.)
  - Coordinate with testing
  - Referee and resolve conflicts